UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/980,761	04/15/2002	Michael R Krause	10003628-2	4254

22879 7590 11/27/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
STRANGE, AARON N	

ART UNIT	PAPER NUMBER
2153	

MAIL DATE	DELIVERY MODE
11/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/980,761
Filing Date: April 15, 2002
Appellant(s): KRAUSE ET AL.

MAILED

NOV 26 2007

Technology Center 2100

Patrick G. Billig
Reg. No. 38,080
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 2/22/07 appealing from the Office action mailed 6/5/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Block	US 6.192,417	Feb. 20, 2001 (filed Mar. 30, 1999)

Request for Comment 793, Transmission Control Protocol (Sept. 1981) (RFC 793).

P.V. Mockapetris, Analysis of Reliable Multicast Algorithms for Local Networks, ACM (1983).

J.M. Aldrich, (USENET post, Oct. 16, 1997).

Request for Comment 2236, Internet Group Management Protocol, Version 2 (Nov. 1997) (RFC 2236).

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 2-18, 22, 25-41, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block).

Regarding claim 2, RFC 793 discloses a distributed computer system comprising:

- a source endnode including:
  a. a source process which produces message data (pg 7, last ¶ continued on pg 8 and pg 24, last ¶, first sentence);
  b. a send work queue having work queue elements that describe the message data for sending (pg 24, last ¶ and pg 41, 3rd ¶, last sentence);
- destination endnode including:
  a. a destination process (pg 7, last ¶ continued on pg 8);
  b. a receive work queue having work queue elements that describe where to place incoming message data (pg 7, last ¶ continued on pg 8);
- communication fabric providing communication between the source endnode and the destination endnode (inherent; pg 7, last ¶ continued on pg 8); and
- an end-to-end context at the source endnode and the destination endnode storing state information to ensure the reception and sequencing of message data sent from the source endnode to the destination endnode thereby permitting reliable datagram service between the source endnode and the destination endnode (Section 2.6 beginning on pg 9).

However, RFC 793 fails to disclose 1) reliable *multicast* to a group of destination endnodes wherein the reliable multicast comprise a series of replicated unicasts to each endnode and 2) a network interface controller having a completion processing unit for generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode.

With regard to point 1, reliable multicasting to a group of recipients through a series of replicated packets was well known in the art at the time of invention, as evidenced by Mockapetris. In a related art Mockapetris teaches a reliable multicasting method where the sender sends a separate (replicated) message to each destination endnode and receives an acknowledgement of receipt from each endnode separately (pg 153 Simulation algorithms, 1st ¶). Mockapetris further discloses that the sender maintains the multicast group of destination endnodes as a list (pg 153 Simulation algorithms, 1st ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to add the multicast functionality disclosed by Mockapetris to the one-to-one transmission system disclosed by RFC 793 in order to provide a straightforward method for reliable one-to-many transmission (pg 153 Simulation algorithms, 1st ¶).

In the combined Mockapetris and RFC 793 system, all one-to-one transmission capabilities defined in the RFC 793 are expanded to a one-to-many (multicast) transmission capability since the one-to-many transmission is simply a series of replicated one-to-one transmissions to each multicast member endnode, as disclosed by Mockapetris (Cited above).

With regard to point 2, while RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received frame (pg 41, 5th ¶) neither RFC 793 nor Mockapetris specifically recited a network interface controller having a completion processing unit for generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode. Nonetheless it was widely known in the art at the time of the invention to notify application source processes that make multicast requests the status of such requests, as evidenced by Block. In a similar multicasting system, Block disclosed multicasting messages to a group of nodes (Col 9, lines 28-34). Like RFC 793 and Mockapetris, Block relies on acknowledgments from each node in the multicast group to determine which nodes have reliably received each multicast message (Col 15, lines 51-57 or Col 16, lines 8-11). In Block's system the source process that submits the multicast request can receive a completion event (notification) when a predetermined percentage of nodes (i.e. all nodes in the multicast group) have reliably received the multicast message (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19). This notification scheme allows the source process that submits a multicast request to be notified of the multicast request status. Additionally Block disclosed that the notification scheme may be implemented using a network interface controller (i.e. the processor and memory of the computing system executing Block's disclosed applications, see inter alia Block Col 8, lines 53-56). Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate

the completion event notification scheme, as disclosed by Block, within the combined RFC 793 and Mockapetris system, so that application processes which use the combined RFC 793 and Mockapetris multicast system are apprised of the status of their multicast requests.

Regarding claim 25, RFC 793 discloses a method of sending message data in a distributed computer system, the method comprising:

- producing message data with a source process at the source endnode (pg 7, last ¶ continued on pg 8 and pg 24, last ¶, first sentence);
- describing the message data for sending with work queue elements in a send work queue at the source endnode (pg 24, last ¶ and pg 41, 3rd ¶, last sentence);
- describing where to place incoming message data with work queue elements in a receive work queue at the destination endnode (pg 7, last ¶ continued on pg 8);
- storing state information in an end-to-end context at the source endnode and the destination endnode to ensure the reception and sequencing of message data sent from the source endnode to the destination endnode (Section 2.6 beginning on pg 9); and
- reliably sending data including performing a unicast of message mdata though the send work queue and the end-to-end context at the source

endnode to the receive work queue and end-to-end context portion at the destination endnodes (Section 2.6 beginning on pg 9).

However, RFC 793 fails to disclose 1) reliable *multicast* to a group of destination endnodes wherein the reliable multicast comprise a series of replicated unicasts to each endnode and 2) generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode.

With regard to point 1, reliable multicasting to a group of recipients through a series of replicated packets was well known in the art at the time of invention, as evidenced by Mockapetris. In a related art Mockapetris teaches a reliable multicasting method where the sender sends a separate (replicated) message to each destination endnode and receives an acknowledgement of receipt from each endnode separately (pg 153 Simulation algorithms, 1st ¶). Mockapetris further discloses that the sender maintains the multicast group of destination endnodes as a list (pg 153 Simulation algorithms, 1st ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to add the multicast functionality disclosed by Mockapetris to the one-to-one transmission system disclosed by RFC 793 in order to provide a straightforward method for reliable one-to-many transmission (pg 153 Simulation algorithms, 1st ¶).

In the combined Mockapetris and RFC 793 system, all one-to-one transmission capabilities defined in the RFC 793 are expanded to a one-to-many (multicast) transmission capability since the one-to-many transmission is simply a series of

replicated one-to-one transmissions to each multicast member endnode, as disclosed by Mockapetris (Cited above).

With regard to point 2, while RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received frame (pg 41, 5th ¶) neither RFC 793 nor Mockapetris specifically recited generating a completion event to the source process in response to an indication that a *predetermined percentage of destination endnodes in the multicast group have reliably received a selected amount of message data multicast* from the source endnode. Nonetheless it was widely known in the art at the time of the invention to notify application source processes that make multicast requests the status of such requests, as evidenced by Block. In a similar multicasting system, Block disclosed multicasting messages to a group of nodes (Col 9, lines 28-34). Like RFC 793 and Mockapetris, Block relies on acknowledgments from each node in the multicast group to determine which nodes have reliably received each multicast message (Col 15, lines 51-57 or Col 16, lines 8-11). In Block's system the source process that submits the multicast request can receive a completion event (notification) when a predetermined percentage of nodes (i.e. all nodes in the multicast group) have reliably received the multicast message (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19). This notification scheme allows the source process that submits a multicast request to be notified of the multicast request status. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the completion event notification scheme, as disclosed by Block, within the combined RFC 793 and Mockapetris system, so that application processes which use

the combined RFC 793 and Mockapetris multicast system are apprised of the status of their multicast requests.

Regarding claims 3 and 26, RFC 793 discloses the source endnode including a network interface controller which packetizes the message data into frames (Section 2.2 beginning on pg 7).

Regarding claims 4 and 27, RFC 793 discloses the system wherein the destination endnodes each include a network interface controller which acknowledges receipt of frames sent from the source endnode (pg 6 Reliability section).

Regarding claims 5 and 28, RFC 793 discloses the system wherein the network interface controller and the end-toend context portion in the destination endnode ensures strong ordering of received frames sent from the source endnode, such that the frames are received in a same defined order as transmitted from the source endnode (pg 6 Reliability section).

Regarding claims 6 and 29, RFC 793 discloses the system wherein the source endnode retransmits frames that are not successively acknowledged in the reliable multicast service (pg 6 Reliability section).

Regarding claims 7 and 30, Mockapetris discloses the system wherein the network interface controller in the source endnode includes hardware which replicates frames to be provided in the series of unicasts (inherent, pg 153 Simulation algorithms, 1st ¶).

Regarding claims 8 and 31, Mockapetris discloses the system wherein the source endnode includes software verbs which perform the series of unicasts as a series of individual sequenced message send operations (pg 153 Simulation algorithms, 1st ¶).

Regarding claims 9 and 32, Mockapetris discloses the system wherein changes in composition of the endnodes participating in the multicast group are communicated to all endnodes participating in the multicast group (inherent, each sender maintains a list of multicast members and each member can be a sender; pg 153 Simulation algorithms, 1st ¶).

Regarding claims 10 and 33, Mockapetris discloses the system wherein the source endnode and each destination endnode maintains a list of destination addresses for all other endnodes participating in the multicast group (pg 153 Simulation algorithms, 1st ¶).

Regarding claims 11-12 and 34-35, RFC 793 discloses generating cumulative and per frame acknowledgments (pg 20, last ¶ continued on to pg 21).

Regarding claims 13-16 and 36-39, RFC 793 discloses gathering and counting acknowledgements from endnodes using a completion processing unit containing a completion queue (retransmission queue) (pg 21, first sentence below Functional Specification heading; pg 22 top ¶). RFC 793 further discloses informing the source process (through TCP-to-user signals) of an operation status of frames (pg 41, 5th ¶).

Regarding claims 17-18 and 40-41, RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received multicast frame (pg 41, 5th ¶). However, both Mockapetris and RFC 793 fail to teach generating a completion event when a certain percentage of endnodes have received the multicast frame. The Examiner takes Office Notice that it was well known in the computer networking art at the time of invention to generate events to a source process based on the percentage of completion (0% to 100%) for a given task. It would have been obvious to one of ordinary skill in the art at the time of invention to generate a completion event when a certain percentage of endnodes received the multicast frame in order to alert the host process of the multicast completion status.

Regarding claims 22 and 45, RFC 793 discloses the system wherein the completion processing unit includes a timing window, wherein expiring of the timing

window without necessary conditions for a completion event for a corresponding multicast frame occurring indicates that any missing acknowledgments art to be tracked and resolved (pg 10, 1st ¶).

Claims 19-21 and 42-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block), as applied to the claims above, and further in view of Aldrich (USNET post, John M. Aldrich Oct 16 1997).

Regarding claims 19 and 42, as discussed above RFC 793 discloses tracking ACKs from each endnode however, RFC 793 fails to teach using a bit-mask array for such tracking. Nevertheless, the use of bit-mask arrays to track events was well known in the art at the time of invention, as evidenced by Aldrich. In a related art, Aldrich discloses using a bit-mask array as a set of flags, which can be set and unset using bitwise operators (pg 2, top ¶). It would have been obvious to one of ordinary skill in the art at the time of invention to use a bit-mask array to track acknowledgements from endnodes in order to minimize memory consumption by only using a single bit to track each acknowledgement.

Regarding claims 20-21 and 43-44, RFC 793 discloses generating a completion event (TCP-to-user signals) for each endnode that acknowledges a received multicast frame (pg 41, 5th ¶). Further as discussed with respect to claims 2 and 25, Block disclosed generating a completion event when a certain percentage of endnodes have received the multicast frame (see inter alia Col 13, lines 40-43 and Col 16, lines 8-11, 18-19).

Claims 23-24 and 46-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Request for Comment 793 (Transmission Control Protocol, hereinafter RFC 793) and Mockapetris (Analysis of Reliable Multicast Algorithms for Local Networks, Paul Mockapetris) and Block et al. (U.S. Patent Number 6,192,417; hereinafter Block), as applied to the claims above, and further in view of Request for Comment 2236 (Internet Group Management Protocol, Version 2; hereinafter RFC 2236).

Regarding claims 23-24 and 46-47, while Mockapetris discusses maintaining a list of multicast members (pg 153 Simulation algorithms, 1st ¶) Mockapetris is silent as to how a multicast member list is updated. In a related art, the Internet Group Management Protocol Version 2 provides a protocol for maintaining multicast group membership (RFC 2236 pg 1, Abstract 2nd ¶) through the use of join (RFC 2236 pg 6, last line) and leave events (RFC 2236 pg 7, 1st bullet). It would have been obvious to one of ordinary skill in the art at the time of invention to incorporate the join and leave

events taught by RFC 2236 within the combined RFC 793 and Mockapetris system in order to allow changes in group membership to be quickly reported to the multicast sender (RFC 2236, Abstract 1st ¶).

(10) Response to Argument

Regarding claims 2-18, 22, 25-41 and 45, of which claims 2 and 25 are independent, Appellants present no separate arguments directed to the dependent claims encompassed by this rejection. Additionally, Appellants present no substantive arguments directed to the separate patentability of dependent claims 19-21, 23, 24, 42-44, 46 and 47, rejected under separate grounds.

The Examiner will address independent claim 2 as representative of all claims on appeal. The rationale given below is equally applicable to Appellants' arguments with respect to claim 25, which substantially mirror those presented for claim 2.

The various points raised by Appellants are summarized below, and each point is addressed individually by the Examiner.

Regarding claim 2, Appellants present three principal arguments:

Appellants' argument a) Block does not teach or suggest "the source endnode participating in the multicast group including a network interface controller having a completion processing unit configured to generate a completion event to the source process in response to an indication that a predetermined percentage of destination

endnodes in the multicast group have reliably received a selected amount of message data multicast from the source node" (Br. 10).

Regarding argument a), the Examiner respectfully disagrees. Block clearly discloses creating a CCMessage object at the source endnode for each message to be sent (col. 16, ll. 4-5). The CCMessage object is kept until a predetermined percentage ("all groups" of 100%) of destination endnodes have reliably received a selected amount of message data (CCMessage is kept until "all nodes and groups that are supposed to receive the message acknowledge receipt")(col. 16, ll. 8-11). The CCMessage further generates a completion event (call "notify object") once the message receipt is completed by all recipients (col. 16, ll. 18-19).

Therefore, Block has a completion processing unit that generates a completion event (call notify object) in response to an indication that a predetermined percentage (100%) of destination endnodes in the multicast group have reliable received a selected amount (all) of message data multicast from the source node.

Appellants' argument b) RFC 793 does not teach or suggest "multiple end-to-end contexts, each end-to-end context having a portion storing state information at the source node and a portion storing state information a corresponding one of the destination endnodes to ensure the reception and sequencing of message data multicast from the source endnode to the corresponding one of the destination endnodes, wherein a reliable multicast comprises a series of replicated unicasts of

message data through the send work queue and each of the end-to-end context portions at the source endnode to the receive work queue and the corresponding end-to-end context portion of each of the destination endnodes" (Br. 10).

Regarding argument b), it is initially noted that Appellants have argued a very large portion of claim 2, which was rejected based on the combination of RFC 793, Mockapetris and Block. With such a general argument, it is difficult to determine exactly what Appellants feel is missing from the combination. Regarding the claimed contexts, RFC 793 was relied upon to teach a conventional unicast end-to-end context, storing state information at both a source and a destination endnode, which ensures the reception and sequencing of message data sent from the source to the destination, permitting reliable datagram service between the nodes. When combined with Mockapetris, which teaches implementing a reliable multicast comprising a series of replicated unicasts of message data, the combination would provide multiple end-to-end contexts, corresponding to each source and destination endnode.

The combined teachings of RFC 793 and Mockapetris, when considered as a whole, teach the entirety of the argued limitation.

Appellants' argument c) Mockapetris does not teach or suggest "reliable multicast comprising a series of replicated unicasts of message data through the send work queue and each of the end-to-end context portions at the source endnode to the

receive work queue and the corresponding end-to-end context portion at each of the destination endnodes" (Br. 12).

Regarding argument c), the Examiner respectfully disagrees. Mockapetris provides a dead on teaching of a reliable multicast comprising "a series of replicated unicasts" from the sender to the destination (p. 153, col. 2, ¶2). As acknowledged by Appellants (Br. 12), the sender transmits separate messages to each destination and receives separate ACKs in return to provide one-to-many multicast transmission. While Appellants generally assert that this differs from the claimed limitation, Appellants have failed specifically pointing out how the language of the claims patentably distinguishes them from the references. The Examiner submits that there is no patentable distinction between the argued limitation and the multicasting method taught by Mockapetris, particularly when considered in context with the combined teachings of RFC 793, Mockapetris and Block.

In summary, each of the cited references relate to network communication, and each limitation of the claim is taught or suggested by the combination of RFC 793, Mockapetris and Block. Appellants have provided only general arguments of patentability, citing large portions of the independent claims, and have failed to specifically point out how the language of the claims patentably distinguishes them from the references. The claims amount to nothing more than the predictable use of familiar

prior art elements according to their established functions. *See KSR Int'l Co. v. Teleflex, Inc.*, 127 S. Ct. 1727, 1740, 82 USPQ2d 1385, 1396 (2007).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Aaron Strange

Conferees:

Lynne H Browne
Appeal Practice Specialist, TQAS
Technology Center 2100

Glenton Burgess
Supervisory Patent Examiner
Art Unit 2153